

Warehouse management system with stacking support

Michał Mrowiec ¹, Tomasz Zając ²

¹ Uniwersytet Bielsko-Bialski, Wydział Budowy Maszyn i Informatyki, Informatyka specjalność: inżynieria oprogramowania, michalmrowiec21@gmail.com

² mgr inż., Uniwersytet Bielsko-Bialski, Wydział Budowy Maszyn i Informatyki, tzajac@ubb.edu.pl

Abstract: The article presents the design and implementation of the author's warehouse management system, which includes support for stacking storage and management of warehouse locations. The first part introduces key concepts related to logistics and storage of goods. Among other things, warehouse processes and goods storage techniques are described. A selection of available solutions is also presented. The next part of the article focuses on the conversion of the realized system. Among other things, the database schema, the adopted concepts of storage units and a fragment of the application interface are discussed.

Keywords: warehouse management system; tiered storage; warehouse locations; logistics;

System magazynowy z obsługą składowania piętrowego

Michał Mrowiec ¹, Tomasz Zając ²

¹ Uniwersytet Bielsko-Bialski, Wydział Budowy Maszyn i Informatyki, Informatyka specjalność: inżynieria oprogramowania, michalmrowiec21@gmail.com

² mgr inż., Uniwersytet Bielsko-Bialski, Wydział Budowy Maszyn i Informatyki, tzajac@ubb.edu.pl

Streszczenie: W artykule przedstawiono projekt i realizację autorskiego systemu zarządzania magazynem, który obejmuje obsługę składowania piętrowego oraz zarządzania lokalizacjami magazynowymi. W pierwszej części przedstawiono kluczowe pojęcia związane z logistyką i magazynowaniem towarów. Opisane zostały m.in. procesy magazynowe oraz techniki składowania towarów. Zaprezentowane zostały również wybrane dostępne rozwiązania. Dalsza część artykułu skupia się na przedstawieniu zrealizowanego systemu. Omówiono m.in. schemat bazodanowych, przyjęte koncepcje jednostek magazynowych oraz zaprezentowano fragment interfejsu aplikacji.

Słowa kluczowe: system zarządzania magazynem; składowanie piętrowe; lokalizacje magazynowe; logistyka;

1. Wprowadzenie

W dzisiejszych czasach dobre zarządzanie magazynem jest kluczowym wyzwaniem dla wielu przedsiębiorstw. Chcąc zoptymalizować koszty oraz przyspieszać procesy logistyczne, istotne staje się maksymalne wykorzystanie przestrzeni magazynowych oraz czasu pracy pracowników i maszyn. W tym celu wykorzystuje się systemy magazynowe Warehouse Management System (WMS), które wspomagają procesy logistyczne w obrębie magazynów. Głównymi funkcjami tego typu aplikacji są obsługa wydań i przyjęć oraz zarządzanie składowaniem. Obecnie szeroko wykorzystywane są różnego typu infrastruktury umożliwiające składowanie piętrowe tak, aby jak najbardziej wykorzystać przestrzeń roboczą. W celu usprawnienia procesu składowania towarów, istotne jest, aby WMS posiadał możliwość precyzyjnej identyfikacji lokalizacji jednostek na terenie magazynu. [1-3]

1.1. Procesy magazynowe i techniki składowania

W celu właściwego zaprojektowania systemu magazynowego, kluczowe jest dogłębne zrozumienie procesów magazynowych oraz technik składowania. Ta część, skupia się na omówieniu głównych aspektów związanych z zarządzaniem magazynem. Opisane zostały operacje logistyczne, metody magazynowania oraz strategię wydawania towarów.

1.1.1. Procesy magazynowe

W ramach procesów magazynowych, wyróżnia się kilka głównych operacji [1,2,4]:

- Przyjmowanie – Podczas tego procesu, do systemu WMS wprowadzane są dane przyjmowanych towarów. Na podstawie tych informacji, system generuje dokument przyjęcia magazynowego. W trakcie procesu, przyjęte towary mogą być również etykietowane, co umożliwia sprawniejsze przeprowadzanie kolejnych operacji logistycznych.
- Składowanie – Polega ono na umieszczeniu towaru w dostępnej przestrzeni magazynowej. Aby zminimalizować ruch towarów na magazynie, system WMS może oferować wybór optymalnego miejsca składowania dla danego towaru, uwzględniając jego parametry oraz rotację. Jednakże, aby to było możliwe, w systemie musi być dokładnie odwzorowana struktura magazynu, w tym wszystkie miejsca składowania. Dla maksymalnego ograniczenia ruchów towarów, stosuje się tak zwany cross-docking. Proces ten polega na przeniesieniu przyjętego towaru, bezpośrednio do miejsca, gdzie jest on potrzeby, z pominięciem procesu składowania.
- Kompletacja – Podczas tej operacji, pracownik ma za zadanie zebrać wszystkie towary z magazynu, według zamówienia odbiorcy. Proces ten może się odbywać w strefie kompletacji lub w strefie składowania. Do realizacji tego procesu, system WMS może zaproponować pracownikowi optymalną ścieżkę pobierania towarów. Operacja kompletacji poprzedza proces wydawania towarów.
- Wydawanie – Proces, w którym skompletowane towary opuszczają magazyn. Podczas realizacji tego procesu, towary są weryfikowane, pakowane, etykietowane i ładowane na środek transportu. Operacja ta powoduje zmniejszenie stanów magazynowych o wydane towary.
- Kontrola stanów magazynowych – Obejmuje ona wyświetlanie stanów magazynowych w systemie oraz inwentaryzację towarów. Inwentaryzacja, inaczej spis z natury, przeprowadza się w celu porównania ilości towarów z systemu, z fizycznym stanem towarów na magazynie. Podczas spisu z natury, przeliczane są wszystkie artykuły podlegające inwentaryzacji oraz wprowadza się korekty do systemu WMS, w przypadku niezgodności.

1.1.2. Techniki składowania

Magazynowane towary, są zwykle grupowane w większe jednostki zwane jednostkami ładunkowymi. Towary w takiej jednostce są zazwyczaj ułożone na nośniku. Najczęściej nośnikiem jest paleta, ale mogą to być też różnego rodzaju skrzynie lub kontenery wielokrotnego użytku. Dzięki grupowaniu towarów w jednostki, mogą być one łatwo przechowywane i transportowane za pomocą np. wózków paletowych różnego typu. [5]

W magazynach wyróżnia się dwie główne techniki składowania [6-8]:

- Składowanie rzędowe – W tej metodzie towary są układane na regałach, wzdłuż jednej ścieżki. Dzięki takiemu ułożeniu, istnieje możliwość swobodnego dostępu do każdego towaru. Systemy regałowe używane w tej technice, to zazwyczaj ramowe konstrukcje o modułowym charakterze. Umożliwia to łączenie ze sobą regałów i wypełnienie obszaru magazynowego w poziomie jak i w pionie. Ze względu na mnogość miejsc, gdzie mogą być składowane towary, ważna jest informacja o dokładnym jego położeniu. Zazwyczaj składowanie rzędowe wykorzystuje się do magazynowania towarów o dużym zróżnicowaniu. Wadą tej techniki jest niski współczynnik wykorzystania powierzchni magazynowej, jednakże dzięki temu podejściu możliwe jest zastosowanie dowolnej strategii wydawania towarów.
- Składowanie blokowe – Ta technika polega na układaniu zasobów bezpośrednio obok siebie i ich piętrowaniu. Jednostki mogą być umieszczane na podłodze lub na specjalnych, przeznaczonych do tego regałach. Istnieje kilka rodzajów konstrukcji do składowania blokowego, są to m.in. regały wjazdowe, przepływowe i satelitarne. Takiego rodzaju magazynowania nie można zastosować do wszystkich rodzajów towarów. Dotyczy to głównie produktów o niskiej odporności na ściskanie i zgniatanie oraz artykułów wrażliwych na wilgotność. W tym podejściu istotny jest również sposób pakowania i ułożenia towarów na jednostkach magazynowych.

Składowanie blokowe jest często wykorzystywane w przypadku strategii wydawania towarów LIFO, które opisano dalej. Takie podejście składowania ma wysoki współczynnik wykorzystania powierzchni magazynowej.

1.1.3. Strategie wydań

W niektórych przypadkach, kluczowa jest odpowiednia strategia zarządzania wydaniem towarów. System WMS powinien uwzględniać niżej wymienione podejścia i proponować odpowiednie towary podczas realizacji operacji logistycznych. Wyróżnia się kilka głównych metod [9]:

- FIFO (First In First Out) – polega ona na wydawaniu w pierwszej kolejności towarów, które były najwcześniej przyjęte na magazyn.
- LIFO (Last In First Out) – w tym podejściu najpierw wydawane są towary, które jako ostatnie weszły na magazyn.
- FEFO (First Expired First Out) – ta metoda polega na wydawaniu w pierwszej kolejności towarów o najkrótszej dacie przydatności. Najczęściej to podejście jest stosowane w przypadku produktów żywnościowych, leków lub kosmetyków, czyli towarów o określonym terminie przydatności.

Odpowiednie planowanie rozmieszczenia towarów w magazynie jest istotnym aspektem efektywnego zarządzania przestrzenią magazynową. W tym celu mogą być stosowane różne algorytmy lub metody, które opierają się na analizach uwzględniających kryteria takie jak częstotliwość rotacji, wartość czy datę przydatności towarów. Zastosowanie optymalnego i racjonalnego układu zasobów, może przyczynić się do lepszego wykorzystania przestrzeni magazynowej oraz redukcji czasu i kosztów. [10]

1.2. Przegląd dostępnych rozwiązań

Na rynku dostępnych jest wiele systemów magazynowych, które są projektowane z myślą o różnych grupach odbiorców. Oprócz realizacji podstawowych operacji logistycznych, mogą one oferować szereg dodatkowych funkcji. Wybrano trzy przykładowe popularne rozwiązania, dostępne na rynku.

1.2.1. Comarch WMS

Comarch WMS jest systemem, który dzieli się na dwie aplikacje, dla osób zarządzających – WMS Zarządzanie oraz dla pracowników magazynu – WMS Magazynier. Oprogramowanie umożliwia zarządzanie magazynem wysokiego składu. Pozwala on m.in. na elastyczne budowanie struktury magazynu, zarządzanie położeniem towarów oraz kontrolowanie stanów magazynowych. Posiada on również mechanizmy AI, która pomaga w pracy magazynu, na przykład w zarządzaniu położeniem towarów. [11]

1.2.2. ExpertWMS Smart

ExpertWMS Smart jest systemem magazynowym firmy DataConsult, który przeznaczony jest głównie dla małych i średnich firm. Oprogramowanie to obejmuje pełną obsługę procesów magazynowych oraz logistycznych. Program umożliwia zarządzanie procesami przyjęć i wydań, różnicując ich możliwe typy oraz realizację ich w różnych jednostkach logicznych, takich jak palety czy towary drobnicowe. Towarom można nadawać dodatkowe cechy takie jak data ważności, numery partii czy kody kreskowe. Moduł odpowiedzialny za magazynowanie oferuje wsparcie dla lokalizacji towarów oraz optymalizacji ich rozmieszczenia na magazynie. Dodatkowo obsługuje pełne i częściowe inwentaryzacje oraz przesunięcia wewnętrzne. [12]

1.2.3. WMS Easyway Qguar

WMS Easyway Qguar jest rozwiązaniem firmy Quantum Qguar i jest przeznaczony dla mniejszych magazynów. Zakres funkcjonalny systemu obejmuje m.in. wielowymiarowy podział miejsc magazynowych, automatyczny i manualny przydział lokacji towarów na magazynie, standardowe operacje logistyczne, w tym inwentaryzacje oraz zarządzanie partiami i datami przydatności. [13]

Powyżej opisane systemy magazynowe implementują możliwość w różnym stopniu zarządzania składowaniem piętrowym oraz lokalizacjami magazynowymi. Oprogramowania te pomimo swoich zalet są rozwiązaniami płatnymi z brakiem dostępu do całego kodu źródłowego.

2. Realizacja systemu magazynowego z obsługą składowania piętrowego

Kluczowym elementem projektowania systemu magazynowego jest precyzyjne określenie wymagań funkcjonalnych, które wspierają codzienne procesy magazynowe. Ważnym aspektem jest również stworzenie odpowiedniego modelu

bazy danych, zapewniającego efektywne zarządzanie danymi. Poniżej przedstawiono główne wymagania funkcjonalne systemu z obsługą składowania piętrowego oraz schemat bazy danych. Na końcu zaprezentowano stworzoną aplikację.

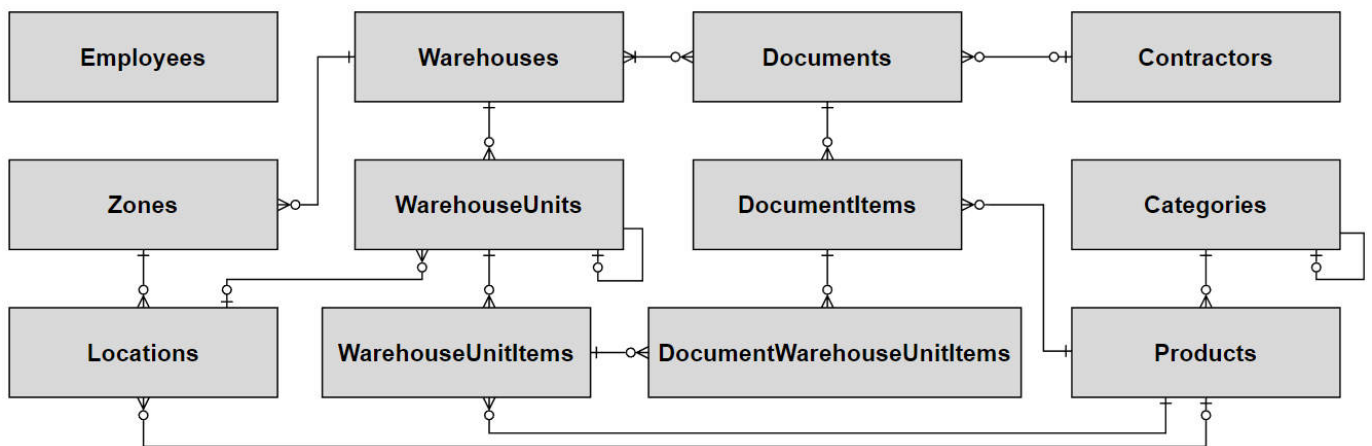
2.1. Wymagania funkcjonalne

Poniżej opisano wymagania funkcjonalne, jakie zostały zaimplementowane w systemie magazynowych z obsługą składowania piętrowego:

- Obsługa dokumentów magazynowych, umożliwia tworzenie, przeglądanie, edytowanie oraz usuwanie dokumentów przyjęć, wydań, przesunięć magazynowych, a także generowanie raportów.
- Obsługa bazy artykułów, zapewnia pełne zarządzanie bazą danych produktów, w tym dodawanie nowych artykułów, edycję istniejących, usuwanie, kategoryzację oraz obsługę kodów kreskowych.
- Obsługa bazy kontrahentów, oferuje możliwość zarządzania kontrahentami magazynu, czyli dodawanie, edycja i przegląd dostawców, klientów oraz innych partnerów biznesowych. System zapewnia przypisywanie kontrahentów do dokumentów magazynowych.
- Przegląd stanów magazynowych, funkcja umożliwia wyświetlanie aktualnych stanów magazynowych.
- Obsługa jednostek magazynowych, system gwarantuje definiowanie jednostek składowania, takich jak palety, skrzynki, kontenery oraz przypisywanie produktów wraz z ilościami do tych jednostek.
- Zabezpieczenia, system implementuje zabezpieczenia, aby dana ilość towaru która jest przypisane do danej operacji logistycznej np. wydania, nie została przypisana do innej operacji.
- Zarządzenie miejscami składowania, system wspiera zarządzanie fizyczną przestrzenią magazynu, czyli lokalizacjami (np. regały, półki, strefy), definiowanie nowych miejsc składowania oraz ich pojemności.
- Zarządzanie piętrowaniem, oprogramowanie umożliwia zarządzanie piętrowaniem jednostek magazynowych (stackowanie), w tym ustalanie, które jednostki mogą być piętrowane. Ważne jest również odzwierciedlenie hierarchii składowania w systemie, czyli informacje o tym, która jednostka magazynowa jest piętrowana na której.
- Zarządzanie użytkownikami systemu, oferuje dodawanie nowych użytkowników, definiowanie ról (np. pracownik magazynu, kierownik, administrator) oraz zarządzanie kontami.

2.2. Schemat bazy danych

Na Rysunku 1 przedstawiono uproszczony schemat bazy danych systemu magazynowego z obsługą składowania piętrowego. Poniżej znajduje się opis poszczególnych tabel i relacji.

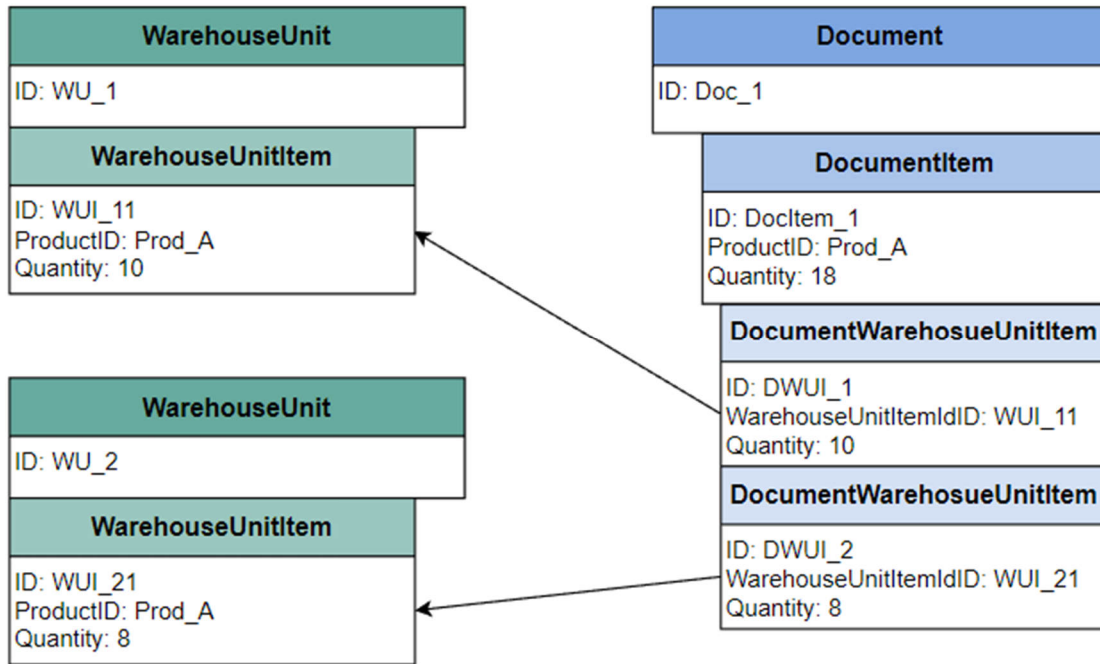


Rysunek 1. Schemat bazy danych

Tabela „Documents” przechowuje dane o dokumentach magazynowych, takich jak przyjęcia, wydania i przesunięcia. Dzięki zastosowaniu Entity Framework, jest ona podzielna na osobne klasy dla poszczególnych typów akcji, dziedziczące ze wspólnej klasy bazowego dokument. Dodatkowo niektóre dokumenty mogą mieć referencje do tabeli „Contractors”, która zawiera dane dostawców oraz odbiorców.

Każdy dokument zawiera pozycje w tabeli „DocumentItems”, znajdują się tam informacje m.in. o produktach, ich ilościach oraz cenach. Pozycje dokumentów posiadają relację z encją „Products”, gdzie znajdują się szczegółowe informacje o artykułach. Jednakże tabela „DocumentItems” również zapisuje podstawowe dane o produktach, z powodu ryzyka, że dane w tabeli z towarami w przyszłości ulegną zmianie.

Tabela „DocumentWarehouseUnitItems” realizuje powiązanie między pozycjami dokumentów, a pozycjami magazynowymi. Dzięki temu, możliwe jest przypisanie jednego artykułu z wielu pozycji magazynowych do jednej pozycji dokumentu w przypadku wydań oraz przydzielenie towaru z pozycji dokumentu do wielu pozycji magazynowych w przypadku przyjęć. Opisany proces przypisania jest zobrazowany na Rysunku 2. Schemat przedstawia dokument mający jedną pozycję, która jest przydzielona do dwóch osobnych jednostek magazynowych, tworząc w nich pozycje magazynowe. Z powodu takiego przypisania, pozycja dokumentu posiada dwa „DocumentWarehouseUnitItem”, które przechowują informacje tym, do których jednostek i w jakich ilości zostały przypisane towary z pozycji dokumentu.



Rysunek 2. Relacja pozycje dokumentu – pozycje magazynowe

System może posiadać wiele magazynów, mieszczących się w tabeli „Warehouses”. Każdy magazyn może zawierać wiele stref składowania, które zdefiniowane są w encji „Zones”. W jednej strefie może znajdować się wiele lokalizacji, za co odpowiedzialna jest tabela „Locations”. Relacja z tabelą „Products”, umożliwiła zdefiniowanie domyślnego nośnika dla danej lokalizacji, ponieważ każdy nośnik, jest produktem.

W każdym magazynie, mogą mieścić się jednostki magazynowe, przechowywane w tabeli „WarehouseUnits”. Rekurencyjne połączenie w tej encji, pozwala na przypisanie klucza głównego jednostki, na której ta jednostka jest piętrzona (stackowana). Dzięki temu możliwe jest dokładne stwierdzenie, która jednostka magazynowa, znajduje się na której.

Encja „WarehouseUnitItem” przechowuje pozycje danych jednostek magazynowych. Dzięki temu, jedna jednostka może posiadać wiele towarów, zawierających różne parametry m.in. datę przydatności, numer partii, wraz z ilością danego produktu. Pozycje magazynowe bezpośrednio odwołują się do tabeli z produktami.

Tabela „Products” przechowuje dane artykułów, a każdy z nich posiada dokładnie jedną kategorię, która zdefiniowana jest w tabeli „Categories”. Każda kategoria może mieć jedną kategorią nadrzędną, co jest realizowane poprzez zastosowanie rekurencyjnego połączenia. Dzięki temu możliwe jest stworzenie hierarchii kategorii, gdzie kategoria nadrzędna ma kilka kategorii podrzędnych.

Użytkownicy systemu, znajdują się w tabeli „Employees”. Każdy użytkownik posiada jedną ze zdefiniowanych ról, czyli administrator, manager lub operator. Jest to podstawowa implementacja, która może zostać rozszerzona.

2.3. Opis architektury proponowanego rozwiązania

System magazynowy został opracowany jako aplikacja webowa. Program składa się dwóch części, aplikacji serwerowej oraz aplikacji klienckiej z interfejsem użytkownika, dostępnego przez przeglądarkę. Takie podejście ułatwia wdrożenie takiego systemu, ponieważ nie wymaga on instalowania oprogramowania na każdej stacji roboczej, a do jego obsługi

wystarczy dowolna przeglądarka internetowa. Dodatkowo umożliwia korzystanie z aplikacji na komputerach jak i na urządzeniach mobilnych, których często używają magazynierzy.

2.3.1. Architektura

Jako architekturę rozwiązania, wybrano tak zwaną czystą architekturę (Clean Architecture). Takie podejście charakteryzuje się podziałem systemu na niezależne od siebie warstwy i luźnymi powiązaniem między nimi [14]. W systemie magazynowych będącego tematem publikacji, wyodrębniono warstwy: interfejsu użytkownika, aplikacji, domeny oraz infrastruktury.

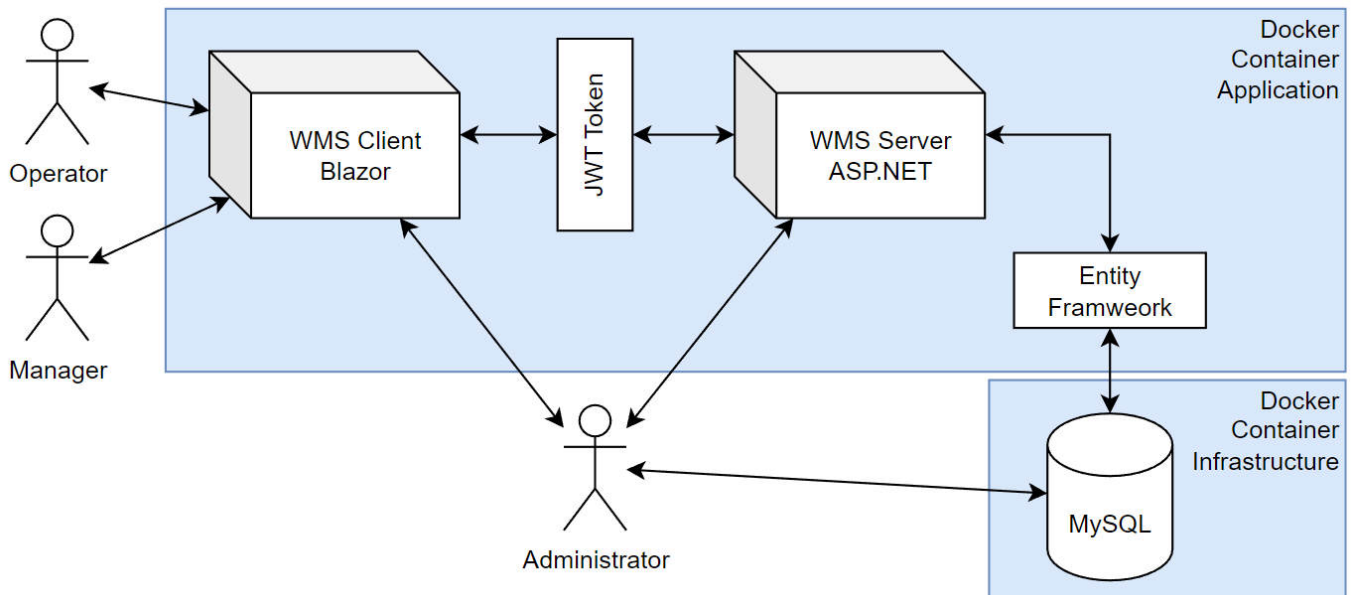
Warstwa domeny, zawiera głównie encje bazodanowe, na podstawie których utworzony został schemat bazy danych. Wraz z domeną, rdzeń systemu tworzy moduł aplikacji, w którym znajduje się cała logika biznesowa. Dodatkowo w rdzeniu programu zdefiniowane są interfejsy, na podstawie których komunikuje się on z innymi warstwami, przez co ta część projektu jest niezależna infrastruktury oraz interfejsu użytkownika.

Warstwa infrastruktury zależna jest od rdzenia aplikacji i jest głównie odpowiedzialna za dostęp do danych. Dzięki zastosowaniu wzorca repozytorium oraz Entity Framework uzyskano niezależność od konkretnego dostawcy bazy danych.

Część odpowiedzialna za interfejs użytkownika dzieli się na projekt API w ASP.NET, będący jednocześnie częścią serwerową Blazor oraz UI w Blazor WebAssembly. API jest zależne od rdzenia aplikacji, korzystając z jego klas i modeli DTO (Data Transfer Object). Projekt Blazor WebAssembly działa wspólnie z serwerem, jednak jest całkowicie niezależny od modeli w pozostałych warstwach systemu, posiadając własne ViewModele. Komunikacja między serwerem a klientem odbywa się poprzez protokół HTTP. Zapytania są mapowane do odpowiednich endpointów w kontrolerach, a dostęp do nich jest autoryzowany poprzez tokeny JWT (Json Web Token).

W celu osiągnięcia luźnego powiązania, w aplikacji wykorzystano wzorzec CQRS (Command-Query Responsibility Segregation). Dzięki temu rozdzielana jest logika tworzenia i modyfikacji danych od pobierania danych, dodatkowo dane operacje posługują się osobnymi modelami. Takie rozwiązanie pozwala na łatwe zarządzanie, skalowalność oraz bezpieczeństwo, co jest kluczowe podczas tworzeniu złożonego systemu. [15]

Do zrealizowania podejścia CQRS zastosowano wzorzec mediator wykorzystując do tego celu bibliotekę MediatR [16]. Mediator zarządza w odpowiedni sposób klasami Command i Query oraz ich Handlerami, przez co odpowiedzialności są odseparowane od siebie, dając możliwość łatwego zarządzania.



Rysunek 3. Schemat architektury systemu

Na Rysunku 3 przedstawiony uproszczony schemat architektury systemu. Widać na nim poszczególne elementy systemu oraz jego aktorów. Cały program działa w dwóch oddzielnych kontenerach Docker, za pomocą Docker Compose, który umożliwia działanie aplikacji wielokontenerowej [17].

2.3.2. Technologie

Poniżej opisano najważniejsze technologie z których korzysta stworzony system magazynowy. Zostały wybrane rozwiązania i technologie nieodpłatne na otwartej licencji, tak aby system mógł być wykorzystywany, bez ponoszenia dodatkowych kosztów za licencje.

Aplikacja została napisana na otwartej źródłowej platformie .NET 8 przy użyciu obiektowego języka C#. Technologia .NET jest wspierana przez firmę Microsoft, oferując niezawodność, bezpieczeństwo oraz wysoką wydajność, a także działanie na wielu systemach operacyjnych m.in. Windows i Linux. Dodatkowo technologia ta udostępnia bibliotekę Blazor, do tworzenia aplikacji webowych. [18]

Do komunikacji i zarządzania bazą danych wybrano Entity Framework Core. Jest to nowoczesne narzędzie typu Object Relational Mapping (ORM), czyli mapper obiektowo-relacyjny. Dzięki użyciu tej biblioteki, eliminowana jest konieczność pisania kodu SQL, ponieważ całe zarządzanie bazą danych odbywa się z poziomu kodu C#. [19]

Jako bazę danych wybrano MySQL. Jest to jedna z najpopularniejszych relacyjnych baz danych, udostępniana na licencji GNU, z dostępem do kodu źródłowego. MySQL jest odpowiednim wyborem dla małych i średnich systemów informatycznych. Zdecydowano się na relacyjną bazę danych, ponieważ procesy magazynowe łatwo przekształcić na wzajemnie powiązane tabele. [20]

Wybrano kontenery Docker jako środowisko działania systemu. To rozwiązanie zapewnia spójną wydajność oraz działanie niezależnie od środowiska, w którym system jest uruchamiany. Tworzenie skonteneryzowanych aplikacji jest obecnie popularnym podejściem, pozwalającym na szybki rozwój i sprawne wdrażanie systemów. [17]

2.3.3. Narzędzia

Podczas pracy nad aplikacją, korzystano głównie z Visual Studio 2022, które jest przeznaczone m.in. do tworzenia aplikacji w języku C# w środowisku .NET. Dodatkowo wykorzystywano Docker Desktop do kontrolowania działania kontenerów, w których działa opracowywany system. W celu kontroli bazy danych, działającą w kontenerze, korzystano z narzędzia Azure Data Studio od firmy Microsoft.

2.4. Testowanie aplikacji

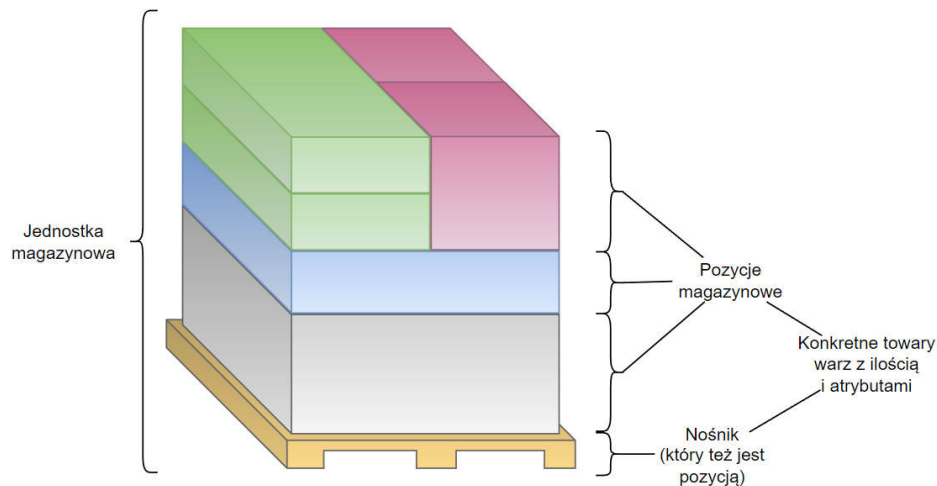
Aby zapewnić najwyższą jakość tworzonego oprogramowania, było ono szeroko testowane na każdym etapie rozwoju. W tym celu wykorzystano testy jednostkowe oraz integracyjne, napisane w bibliotece xUnit. Ponadto endpointy API były na bieżąco testowane za pomocą narzędzia Swagger oraz Postman. Dodatkowo poprawność wszystkich funkcji dostępnych w interfejsie użytkownika zostały przetestowane manualnie.

3. Rezultaty

W tej części, przedstawione zostały rezultaty zaprojektowanego systemu magazynowego z obsługą składowania piętrowego. Na początku przedstawiono koncepcje składowania towarów oraz sposoby w jak może to być realizowane. Dalsza część rozdziału prezentuje interfejs użytkownika stworzonego systemu, udostępniając informacje o zasobach magazynu.

3.1. Realizacja koncepcji magazynowania

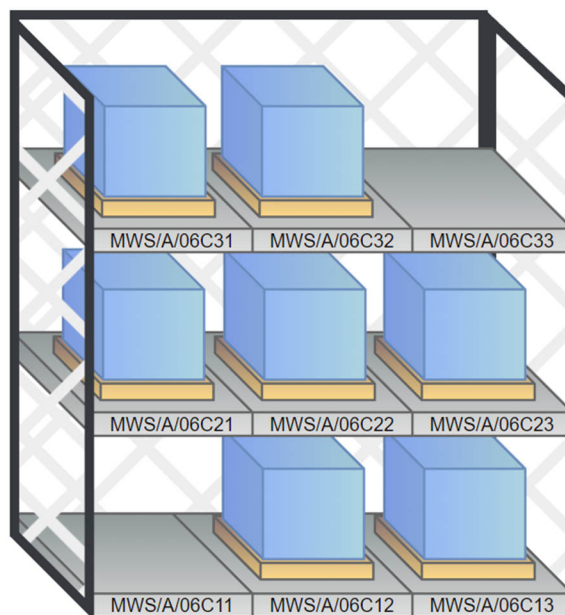
Na Rysunku 4 przedstawiono koncepcję jednostek magazynowych w stworzonym systemie magazynowym. Jako główną jednostkę w systemie rozumie się jednostkę magazynową, która to odpowiada towarom ułożonym na danym nośniku (np. palecie czy skrzyni) oraz dodatkowo zdefiniowane ma atrybuty takie jak możliwość piętrowania, wymiary, wagę oraz temperaturę przechowywania.



Rysunek 4. Jednostka magazynowa

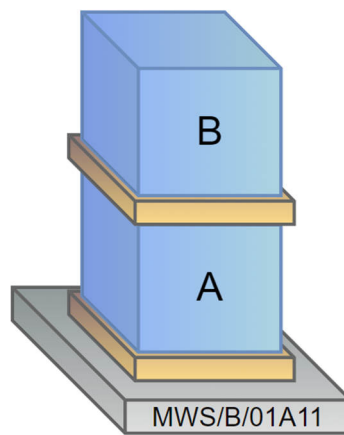
Każda jednostka magazynowa posiada pozycje magazynowe. Pozycje jednostek magazynowych posiadają informacje takie jak ID artykułu, ilość oraz ilość zablokowaną, numer partii, datę przydatności oraz flagę definiującą czy dana pozycja jest nośnikiem. Dzięki temu, w każdej pojedynczej jednostce mogą znajdować się różne towary, o różnych parametrach.

Informacje o zablokowanej ilości danej pozycji magazynowej, służą zabezpieczeniu przed przypisaniem tego samego towaru, do wielu operacji jednocześnie. W przypadku dodawania pozycji do np. wydań magazynowych, możliwe jest dodanie tylko ilości, która pozostaje niezablokowana. Dodatkowo sama jednostka magazynowa również posiada flagę blokady, która jest wykorzystywana podczas przesunięć magazynowych.



Rysunek 5. Składowanie piętrowe na regale

Na Rysunku 5 przedstawiono koncepcję składowania rzędowego, w którym zapasy magazynowe są piętrowane na specjalnych regałach. Każda komórka danego regału ma swój unikalny kod, skonstruowany w taki sposób, aby można było daną lokalizację łatwo odnaleźć w przestrzeni magazynowej. Przyjęto następujący format unikalnego identyfikatora lokalizacji: WWW/Z/RRCLC. Składa się on z identyfikatora magazynu (WWW), identyfikatora strefy w danym magazynie (Z), numeru regału (RR), kodu kolumny będącego literą (C), poziomem (L) oraz komórki (C). Ponadto każda lokalizacja ma określoną pojemność.



Rysunek 6. Piętrzenie jednostek

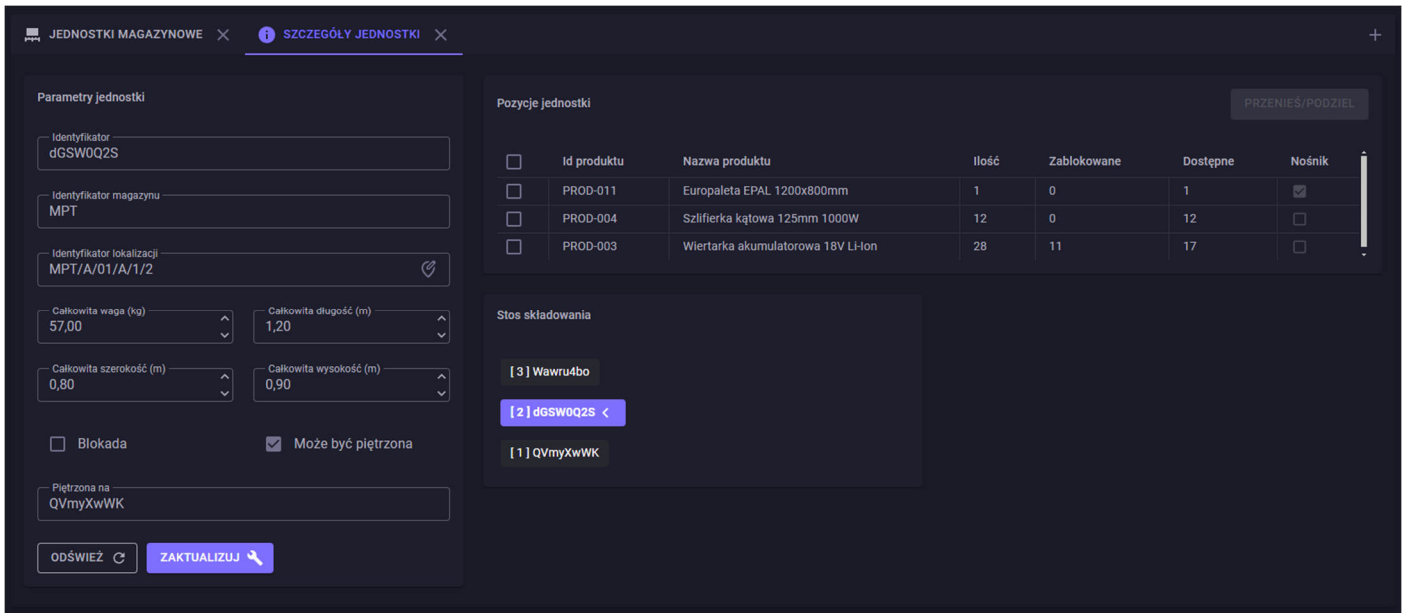
Rysunek 6 obrazuje koncepcję składowania blokowego, piętrząc jednostki bezpośrednio jedna na drugiej. To czy dana jednostka może być piętrzona, definiuje odpowiednia flaga we właściwościach jednostek magazynowych. Aby określić, która jednostka jest ułożona na której, jednostki piętrzone zawierają odniesienie do tej znajdującej się poniżej, zapisane jako jej identyfikator. Odnosząc się do rysunku, to obie jednostki muszą mieć ustawioną flagę definiującą, że mogą być piętrwane, a jednostka B, posiada zapisaną informację, że jest piętrwana na jednostce A. Taki stos również może być umieszczony w konkretnej lokalizacji, wtedy obie jednostki posiadają ten sam identyfikator lokalizacji.

3.2. Projekt zrealizowanej aplikacji

Na poniższych rysunkach przedstawiono widoki interfejsu użytkownika zrealizowanego systemu magazynowego.

Rysunek 7. Widok listy jednostek magazynowych

Rysunek 7 przedstawia widok listy jednostek magazynowych wraz z ich pozycjami. Jest to główny widok, udostępniający wiele informacji na temat zapasów magazynu oraz jego struktury. W tym panelu możliwe jest m.in. filtrowanie i sortowanie jednostek oraz ustawianie lokalizacji, bądź piętrwania. Ważną funkcją jest podgląd zawartości jednostki i wyświetlanie listy pozycji magazynowych, dzięki czemu możliwe jest szybkie stwierdzenie co zawiera dana jednostka bez wchodzenia w szczegóły.



Rysunek 8. Zakładka ze szczegółami jednostki magazynowej

Na Rysunku 8 zaprezentowany został panel szczegółów jednostki magazynowej. Ten widok dzieli się na kilka sekcji, prezentując różne informacje. Poza funkcją informacyjną, pozawala on również na dokonanie dalszych działań, związanych z daną jednostką.

W głównym panelu po lewej stronie, widoczne są wszystkie najważniejsze parametry jednostki, umożliwiając ich edycję. Po prawej stronie, na górze znajduje się lista pozycji magazynowych, które są przydzielone do przeglądanej jednostki. Zaznaczając dowolne pozycje w liście, przycisk „PRZENIEŚ/PODZIEL” staje się aktywny i daje możliwość przeniesienia całości bądź części pozycji, tworząc nowe pozycje w wybranej innej jednostce.

Sekcja „Stos składowania” daje pogląd w jaki sposób jednostki są piętrzone, udostępniając identyfikatory tych jednostek i umożliwiając kliknięcie i podejrzeń ich szczegółów. Aktualnie przeglądana jednostka jest wyróżniona w prezentowanym stosie, dzięki czemu dokładnie widać, że jest to jednostka w środku, umieszczona na innej jednostce, mająca kolejną na sobie.

4. Kierunki dalszego rozwoju

Oprogramowanie ma wiele potencjalnych kierunków rozwoju, począwszy od optymalizacji, przez integracje zewnętrznymi systemami, po dodawanie nowych funkcjonalności oraz obsługi specjalistycznych urządzeń magazynowych.

Istnieje możliwość stworzenia wizualizacji struktury magazynu ze strefami i lokalizacjami, która będzie dostępna w interfejsie użytkownika. Przyczyni się to do łatwiejszego zrozumienia układu magazynu oraz będzie można łatwo zlokalizować dane jednostki. Dodatkowo można zaimplementować obliczanie ścieżki kompletacji i automatyczny dobór pozycji podczas operacji magazynowych, uwzględniając czas i odległości.

Warto opracować moduł planowania i zarządzania przebiegiem inwentaryzacji. Przykładowa funkcjonalność mogłaby polegać na tworzeniu remanentu tylko dla danej rodziny kategorii lub tylko konkretnych artykułów. Wyniki spisu z natury powinny być przedstawiane w przejrzystym panelu podsumowania.

Oczywistym kierunkiem rozwoju jest dodanie integracji opisywanego systemu z innymi systemami planowania zasobów przedsiębiorstwa, czyli Enterprise Resource Planning (ERP). Poza systemami ERP, wartościowa byłaby również integracja aplikacji magazynowej z systemami wspierającymi transport i logistykę określanymi jako Transport Management System (TMS). Jednakże możliwości integracji z zewnętrznymi źródłami są dużo większe.

5. Podsumowania

W artykule przedstawiono projekt i realizację systemu magazynowego z obsługą składowania piętrowego, który odpowiada na dzisiejsze potrzeby zarządzania przestrzenią składowania. Opisane zostały również procesy i techniki magazynowania, które zostały zaimplementowane w systemie. Ponadto wskazano wiele kierunków dalszego rozwoju aplikacji, czemu sprzyja otwarty kod programu.

Wykorzystanie popularnego języka programowania C# oraz nowoczesnych technologii, takich jak Blazor i Entity Framework, ułatwia skalowanie i dostosowywanie systemu do specyficznych wymagań użytkowników. Zastosowana czysta architektura wspiera dalszy rozwój oprogramowania oraz jego integrację z innymi systemami. Dodatkowo, opracowanie aplikacji jako systemu działającego w kontenerach Docker zapewnia łatwość rozwoju, a także umożliwia szybkie wdrażanie i aktualizacje.

Stworzony system pozwala nie tylko zoptymalizować przestrzeń magazynową, ale także może przyczynić się do poprawy efektywności operacyjnej, redukując czas potrzebny na kompletację zamówień. Dzięki dokładnym informacjom o lokalizacjach i sposobie ułożenia jednostek, pracownicy magazynu mogą szybciej i skuteczniej odnajdywać poszczególne towary.

Co więcej, system jest dostępny jako darmowe i otwarte źródłowe oprogramowanie, co umożliwia jego swobodne modyfikowanie i dostosowywanie do specyficznych potrzeb użytkowników, bez dodatkowych kosztów licencyjnych. Dzięki temu każda firma może skorzystać z rozwiązań zwiększających wydajność i efektywność swoich operacji magazynowych, niezależnie od skali działalności.

Literatura

1. Mecalux S.A. Co to jest WMS? Dostępne online: <https://www.mecalux.pl/podrecznik-magazynowania/magazyn/co-to-jest-wms> (data uzyskania dostępu: 15 października 2024).
2. Urbas A., Czech P., Barcik J. Rola i znaczenie zarządzania informatycznego w magazynie. *Zeszyty Naukowe. Transport/Politechnika Śląska* 2011, s. 93-100.
3. Zhang, J.; Onal, S.; Das, S. The dynamic stocking location problem – Dispersing inventory in fulfillment warehouses with explosive storage. *International Journal of Production Economics* 2020, 224, 107550.
4. Jerzyło P., Rutkowska N., Wawrzyńska A. Zarządzanie procesami logistyki magazynowej. *Scientific Journal of Gdynia Maritime University* 2016, s. 37-54.
5. Mecalux S.A. Co to jest jednostka ładunkowa? Dostępne online: <https://www.mecalux.pl/podrecznik-magazynowania/magazyn/jednostka-ladunkowa> (data uzyskania dostępu: 7 października 2024).
6. Darlog Kiedy wybrać składowanie blokowe, a kiedy składowanie rzędowe? Dostępne online: <https://www.darlog.pl/baza-wiedzy/organizacja-magazynu/kiedy-wybrac-skladowanie-blokowe-a-kiedy-skladowanie-rzedowe/> (data uzyskania dostępu: 15 października 2024).
7. WDX S.A. Składowanie rzędowe, a składowanie blokowe. Dostępne online: <https://wdx.pl/2020/03/17/skladowanie-rzedowe-a-skladowanie-blokowe/> (data uzyskania dostępu: 15 października 2024).
8. "WARAN" K.M.A.G. Żebrakowscy Sp. jawna. Sposoby składowania towarów w magazynie. Dostępne online: <https://www.waran-regaly.pl/aktualnosci/sposoby-skladowania-towarow-w-magazynie-art122/> (data uzyskania dostępu: 7 października 2024).
9. Jungheinrich AG. Systemy magazynowania – metody składowania towarów w magazynie. Dostępne online: <https://www.jungheinrich.pl/o-nas/artykuly-prasowe-i-wydarzenia/blog/systemy-magazynowania-metody-sk%C5%82adowania-towar%C3%B3w-w-magazynie-890468> (data uzyskania dostępu: 7 października 2024).
10. Lichota G. Proces magazynowania. Dostępne online: https://mfiles.pl/pl/index.php/Proces_magazynowania (data uzyskania dostępu: 15 października 2024).
11. Comarch SA. Comarch WMS. Dostępne online: <https://www.comarch.pl/erp/wms/> (data uzyskania dostępu: 15 października 2024).
12. DataConsult. ExpertWMS Smart. Dostępne online: <https://dataconsult.pl/expertwms-smart/> (data uzyskania dostępu: 15 października 2024).
13. Quantum Qguar. System WMS EasyWay Qguar. Dostępne online: <https://quantum-software.com/system-wms-easyway-qguar/> (data uzyskania dostępu: 15 października 2024).
14. Martin, R.C. Czysta architektura. Struktura i design oprogramowania. *Przewodnik dla profesjonalistów*; Helion: Gliwice, Polska, 2018.
15. Microsoft. Wzorzec CQRS. Dostępne online: <https://learn.microsoft.com/pl-pl/azure/architecture/patterns/cqrs> (data uzyskania dostępu: 9 października 2024).
16. Bogard J. MediatR. Dostępne online: <https://github.com/jbogard/MediatR> (data uzyskania dostępu: 9 października 2024).
17. Docker Inc. Docker. Dostępne online: www.docker.com (data uzyskania dostępu: 10 października 2024).

18. Microsoft. Build it with .NET. Microsoft 2024; dostępne online: <https://dotnet.microsoft.com/en-us/> (data uzyskania dostępu: 15 października 2024).
19. Microsoft. Centrum dokumentacji programu Entity Framework. Dostępne online: <https://learn.microsoft.com/pl-pl/ef/> (data uzyskania dostępu: 15 października 2024).
20. Kroc K., Kizun O., Skublewska-Paszkowska M. Analiza porównawcza wydajności relacyjnych baz danych MySQL, PostgreSQL, MariaDB oraz H2. *Journal of Computer Sciences Institute* 2020, s. 1-7.